By Express Mail # EL489597185US

APPLICATION FOR UNITED STATES LETTERS PATENT

CRASH RECOVERY SYSTEM

Inventors:
Gunnar ÄIJÄ
Alexander LARSSON

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the installation of software and recovering from installation and runtime errors in a continuously updated and extended distributed computer environment.

2. Description of the Related Art

Many software systems are continuously being updated and extended with new system components and applications. Therefore, purchasers of such software systems often receive updates from the software manufacturers to update the original versions. These updates are typically transmitted by a server of the software provider to the client via a communication network such as the internet. The updates may be automatically sent or a user may download a new version of software via the internet. If any errors occur during either the download, installation, or runtime of the software, it is desirable if the client device can recover from the error. Preferably, the client device reverts back to a previous version of the software system. Furthermore, the installation itself, especially if it is automatic, must be performed so that it does not disturb the user's use of the device.

PCT Patent Publication No. WO 00/58834 discloses a software installation and recovery system for a television viewing computer system. In that system, the computer system has a three stage startup procedure including a bootloader stage, an operating system kernel stage, and an application stage. A persistent storage (hard disk) has two partitions for each stage of the three stage startup procedure, a primary partition and a backup partition. More specifically, there are two partitions dedicated to holding a copy of the second stage boot loader, two partitions holding a copy of the operating system kernel, and two partitions holding a copy of the application software. An indication is recorded that specifies which of the partitions of each pair is the primary and which is the backup. When a new software image is installed, the new image is first copied into the appropriate backup partition and an indication is made in the database that the software installation is underway. The primary and backup partitions are then swapped and the system is rebooted using the new primary partition of the software that has just been downloaded. Control is passed to the backup partition for each software component if the primary partition fails to load properly.

This prior art device includes a separate backup partition for the bootloader stage, the operating kernel loading stage, and the applications loading stage of the installation procedure. The disclosed system detects only failures during loading of programs associated with these stages. That is, this system detects whether the loading process for each of these stages is successfully completed. Accordingly, this system does not address runtime failures, i.e., failures which occur during the running of the programs such as if the program becomes non-responsive or "hangs".

SUMMARY OF THE INVENTION

The object of the present invention is to provide a client device and a method for keeping a software system that is continuously updated and extended consistent and stable over time.

The object is met by a method for recovering from installation and runtime errors on a client device, wherein the client device includes a persistent memory such as a hard drive including first and second system partitions and a package partition. The runtime components of a software system of the client device are installed in the system partitions. All installed packages are saved in the package partition. Furthermore, one of the first and second system partitions is designated the "current" partition and the other is designated the "back-up partition". The dual system partitions are used to perform a recovery procedure in which the software system returns to a previous version in response to a startup (boot) failure, a runtime failure (i.e. a failure that causes the program to “hang”), or a user request. Since at least all the software packages installed in the computer that have been downloaded from a remote server are saved on the package partition, it is possible to selectively reinstall these system components such as the operating system kernel and applications. If the software packages for all the installed system components are saved on the package partition, it is also possible to fully reinstall the entire software system by installing all the software packages contained on the package partition.

When a new software package is received by the client device, the new software package is installed in one of the system partitions and is saved in the package partition. If the software package contains a critical component of the software system, i.e., a component that is continuously loaded or continuously running, the current system partition is first copied onto the backup partition. Then the new software package is installed on the backup system partition and saved on the package partition. If the new software package is a new version of an existing package, then the old version is deleted from the package partition. The designations of the backup and current system partitions are switched and the software system is restarted (rebooted) using the new current system partition which includes the new software. If a booting failure occurs or a runtime failure occurs, the recovery procedure described above is entered.

If the new software package does not contain a critical component, the new software package is installed on the current system partition and saved to the package partition. If the new software package includes a new version of an existing package, then the old version is deleted from the package partition. If a booting failure occurs or a runtime failure occurs, the recovery procedure described above is entered. The recovery procedure may also be initiated by a user of the client device to revert back to a previous version of the software system.

The above-described installation process and recovery process are particularly useful when the client device is a "smart accessory" such as a mobile phone, a set top box for cable or satellite television services, a personal digital assistant (PDA), or a personal computer (PC) which automatically receives software updates from a service provider. If a loading failure or runtime failure occurs, the smart accessory will revert to a previous version of the software. The user of the smart accessory can continue using the client device. Furthermore, the service provider may then provide a further update packet for non-user-interactive (automatic) installation to cure a bug or security problem.

The object of the invention is also met by a device having a persistent memory including first and second system partitions and a package partition for implementing the above described method.

Other objects and features of the present invention will become apparent from the following detailed description considered in conjunction with the accompanying drawings. It is to be understood, however, that the drawings are designed solely for purposes of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims. It should be further understood that the drawings are not necessarily drawn to scale and that, unless otherwise indicated, they are merely intended to conceptually illustrate the structures and procedures described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings, wherein like reference characters denote similar elements throughout the several views:

Fig. 1 is a schematic diagram of system in which the client device according to the present invention is connected;

Fig. 2 is a partial schematic diagram showing portions of the client device according to the present invention;

Fig. 3 is a flow diagram showing the method according to the present invention of installing a new software package;

Fig. 4 is a flow diagram showing an upgrade procedure of the method in Fig. 3;

Fig. 5 is a flow diagram showing an update procedure of the method of Fig. 3;

Fig. 6 is a flow diagram showing a recovery procedure for the client device of the present invention; and

Fig. 7 is a flow diagram showing use of a packet management system.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

Fig. 1 shows a distributed computer environment in which client devices 100, 101, 101, and 102 according to the present invention may be arranged. Each of the client devices 100 - 102 is connectable to servers 300 and 301 via a communication network 200 such as the world wide web or a satellite broadcast network. To maintain software or to install new software on one of the client devices 100, the server 300 may download software packages to the client device 100 through the communication network 200. The downloading may occur automatically or may be initiated by a user of the client device 100, i.e., the user may request a copy of a new version of the software. Although four client devices and two servers are shown, any number of devices may be connected through the network. Furthermore, the server 300 may be arranged so that the downloading of the software package occurs via a broadcast transmission by the server 300 to all client devices to which the software package applies.

Each of the client devices 100 - 102 may be any type of device that is connectable to a remote server through a communication network such as, for example, a mobile phone, a set top box for cable or satellite television services, a personal digital assistant (PDA), or a personal computer (PC).

Fig. 2 is a partial schematic diagram of the client device 100 which includes a persistent memory 10 such as a hard drive or other non-volatile memory and a CPU 20. The persistent memory has first and second system partitions 12, 14 and a package partition 16. The first and second system partitions 12, 14 are collectively referred to as the system partitions 12, 14.

During operation, one of the system partitions 12, 14 is designated the "current" system partition and the other is designated the "backup" system partition. This designation is stored in a boot indicator 24 in a non-volatile memory 26 of the client device 100. The non-volatile memory 26 may comprise a part of the persistent memory 10 or may comprise a separate non-volatile memory. When the client device 100 starts, i.e., boots, the "current" one of the system partitions 12, 14 is used. The information in the boot indicator 24 instructs the CPU 20 regarding which of the system partitions 12, 14 is the current partition.

Runtime components of the software system of the client device 100 are stored in the system partitions 12, 14. The runtime components include an operating system kernel, system commands, configuration files, libraries, a window system, applications and tools, and the user interface. When no new software package is being downloaded to the client device 100, the current system partition contains the latest version of the runtime components of the software system and the backup system partition contains the previous version of the runtime components of the software system. The system partitions 12, 14 are used for the maintenance and installation of software and for recovering from a "crash" of the client device 100 by allowing an automatic return to the previous version of the system software in response to the crash as discussed in further detail below.

The package partition 16 is a space in the persistent memory 10 which may include a copy of all of the software packages installed on the computer (thus providing a means for reinstalling the software system from scratch). Alternatively, the package partition 16 may include copies of only the new downloaded software packages. A package management system 18 includes a package database in each of the system partitions 10, 12 including information relating to the dependencies between the software packages and versioning information about the software packages installed in the associated ones of the system partitions 10, 12.

Fig. 3 shows the process for installing new software packages onto the client device 100. For the description of the installation of a new software package, we will assume that the first system partition 12 is the current partition and the second system partition 14 is the backup partition at the start of the process for installing new software packages.

When a software installation package is received by the client device 100 from a server 300, step S10, the client device 100 must determine whether the software installation package includes a critical component of the system, step S12. To accomplish this, the software package is transmitted from the server 300 with a meta-file which includes specific information regarding whether the software component package includes a critical component. If the software does contain a critical component, an upgrade is performed in step S14. If the software contains only non-critical components, an update is performed in step S16.

Critical components are components that are continuously running or loaded and that require a system reboot/restart to run when upgraded. In the present embodiment, e.g., the operating system kernel, the device drivers, and the window system are considered critical components. Non-critical components include, e.g., new applications, system commands, and configurations files. Other components must be evaluated by the software manufacturer on a case by case basis to determine whether they are considered critical components.

Fig. 4 illustrates the steps for the upgrade procedure which applies to the installation of a software package containing a critical software component. In step S20, an indication is made in the non-volatile memory of the client device 100 that the upgrade procedure has started. This indication allows the client device 100 to determine whether the upgrade was interrupted before being finished. Then the current system partition, i.e., the first system partition 12 in this case, is copied onto the backup system partition, i.e., the second system partition 14, step S22. The new software package is then installed on the backup system partition 14, step S24. The new software package is then also saved on the package partition 16, step S26. If the new software package is a replacement of a preexisting version, then the preexisting version of the software package is deleted from the package partition, step S28. In addition, the information in the package management system 18 must be updated for the new software package, step S29. This information may be in a meta-file sent with the new software package. After the new software package is installed, the boot indicator 24 in the non-volatile memory 26 is updated so that the backup system partition, i.e., the second system partition, is labeled as the current system partition and the current system partition, the first system partition, is labeled as the backup system partition, step S30. Accordingly, after step S30, the second system partition 14 is designated as the current system partition and the first system partition 12 is designated as the backup system partition. In step S32, the indication in the nonvolatile memory 26 is changed to indicate that the upgrade procedure is finished. The client device 100 is now rebooted in step S34 using the current system partition 14 which includes the new software package.

Fig. 5 shows the steps for the update procedure which applies to the installation of a software package containing only non-critical software components. In step S40, an indication is made in a non-volatile memory of the client device 100 that the update procedure has started. This indication allows the client device 100 to determine whether the update procedure was interrupted before being finished. The new software package is then installed into the current system partition 12, step S42. The new software package is then also saved on the package partition 16, step S44. If the new software package is a replacement of a preexisting version, then the preexisting version of the software package is deleted from the package partition, step S46. The information in the package management system 18 is updated for the new software package, step S47. As stated above, this information may be in the meta-file sent with the new software package. In step S48, the indication in the nonvolatile memory is changed to indicate that the update procedure is finished. A reboot is not needed for installation of non-critical components.

If a failure, i.e., a system crash, occurs in the client device 100 after the installation process of Fig. 3, the client device initiates a recovery process. Fig. 6 is a flow diagram illustrating the recovery process. The recovery process may be initiated in step S60 by a startup (boot) failure, a runtime (hangup) failure, or by a user request to revert back to a previous version of the system. Instead of initiating the recovery period at each runtime failure, the recovery process may alternatively be initiated after a number of hangups, i.e., five, occur within a predefined time period. After initiation of the recovery process, the client device 100 starts, i.e., boots up, from the backup one of the first and second system partitions 12, 14, step S62. If the upgrade procedure of Fig. 4 has just been performed the backup system partition is the first system partition 12, and if the update procedure of Fig. 5 has been performed the backup system partition is the second system partition. The boot indicator 24 of the non-volatile memory 26 is then updated to indicate that the backup system partition is now the new current system partition, step S64. The switching of the boot indicator may be performed before rebooting of step S62.

A new backup system partition is then created by copying the previously current system partition, step S66, i.e., copying the system partition that was current during the step S60. The client device 100 then presents to the user a list of all software packages that were installed on the previously current system partition, step S68. This list of software packages may be kept on or derived from package databases of the package management system 18, which are updated for each upgrade and update procedure. The user may then selectively install application and system components on the system partition, step S70, from the list. Accordingly, if a user determines that a component that was previously installed caused the system to crash, the user can remove that component from the system package to avoid further problems in step S70.

Alternatively, the user could completely reinstall the system of the user device 100 in step S70 by installing all software packages saved on the package partition.

Accordingly, if the client device 100 is a mobile phone or a set top box for a cable or satellite television system, the user of the client device can continue using the device after a system crashes caused by an automatic download of a software package from the server 300. In this case, the user at the server 300 can selectively install the software programs as required.

Fig. 7 is a flow diagram showing how the package management system 18 may be used according to the present invention during the step S70 of Fig. 6. The package management system 18 provides information to the user regarding the dependencies between the various software packages on the package partition. When a user selects a user selected software package for installation in the system partitions during the recovery procedure after a "crash", step S80, the package management system 18 checks for compatibility with the existing software packages, step S82. If the user selected software package is not compatible with one of the existing software packages, the CPU 20 issues a warning and/or blocks the installation of the user selected software package, step S84.

The package management system 18 then determines whether the user selected software package requires any further software packages to support the user selected software package, step S86. If the user selected software package requires further software packages, the CPU 20 issues a warning and/or automatically includes the required software packages with the installation of the user selected software package, step 88.

Thus, while there have shown and described and pointed out fundamental novel features of the invention as applied to a preferred embodiment thereof, it will be understood that various omissions and substitutions and changes in the form and details of the devices illustrated, and in their operation, may be made by those skilled in the art without departing from the spirit of the invention. For example, it is expressly intended that all combinations of those elements and/or method steps which perform substantially the same function in substantially the same way to achieve the same results are within the scope of the invention. Moreover, it should be recognized that structures and/or elements and/or method steps shown and/or described in connection with any disclosed form or embodiment of the invention may be incorporated in any other disclosed or described or suggested form or embodiment as a general matter of design choice. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto.